



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3745 — Testing
2022 - 2

Tarea 2

Fecha de entrega: Miércoles 31 de Agosto del 2022 a las 23:59

Información General

La siguiente tarea contempla como objetivo principal crear: reglas de detección de code-smells y reglas de transformación para programas escritos en Python. Varias de las practicas de testing vistas en las clases teóricas se pueden realizar de dos formas: automática y manual. El análisis estático nos permite realizar ciertos tipos de pruebas de forma automática.

Objetivos

- Entender en detalle como las herramientas de análisis estático están implementadas.
- Crear reglas de revisión de código fuente automática y personalizada.
- Crear transformaciones automática de código fuente, útiles para auto-reparar defectos.

Contexto

En clase se vio el módulo `ast`¹ de python que modela la sintaxis de python y ayuda a analizar código escrito en python a través de algoritmos basados en árboles de sintaxis abstracta. Asimismo, se creó una pequeña librería que permite detectar/reportar “warnings” y ejecutar transformaciones de código automáticamente.

Tabla 1. Considere la siguiente lista de warnings:

Nombre	Ejemplo	Descripción
<code>SuspiciousVariableName</code>	<pre>a, b, c, ... self.a self.b ...</pre>	Se mostrará un warning en consola por cada variable cuyo nombre tenga solo una letra. Se debe considerar dos casos: variables temporales y variables de instancia.
<code>NeverReadedVariable</code>	<pre>def example(): x = 2 y = 3 z = 4 return y + z</pre>	Se debe mostrar un warning en consola por cada variable temporal inicializada pero no usada dentro de un mismo método. En el ejemplo, la variable <code>x</code> no está siendo usada.
<code>DataClass</code>	<pre>class Person: def __init__(self): self.name def getName(self): return self.name def setName(self, newName): self.name = newName</pre>	Se mostrará un warning en consola por cada data class en el código. Una data class es una clase que solo tiene atributos y accedores.

Tabla 2. Considere la siguiente lista de transformaciones:

Nombre	Antes	Después
<code>PlusPlus</code>	<pre>x = x + 1</pre>	<pre>x += 1</pre>
<code>IfWithoutElse</code>	<pre>... if <expresions>: <if-body> else: pass ...</pre>	<pre>... if <expresions>: <if-body> ...</pre>

¹<https://docs.python.org/3/library/ast.html>

Tarea

Desarrolle las siguientes actividades:

- **Actividad 1** – Implementar tres nuevas reglas (Rule) que reporten los warnings: `SuspiciousVariableName` (1 punto), `NeverReadedVariable` (1 punto), `DataClass` (2 puntos).
- **Actividad 2** – Implementar dos nuevos transformadores de código (Rewriter) que ejecuten las transformaciones `PlusPlus` (1 punto) y `IfWithoutElse` (1 punto)
- **Actividad 3 (Bonus)** – Proponga e implemente una regla o transformación que sea útil para ayudar al los profesores de Introducción a la Programación. Se publicarán los videos en canvas y se dará un google form a sus compañeros del curso para votación. El primer lugar tendrá +0.2 décimas y el segundo lugar +0.1 décimas. Las décimas irán a la evaluación donde el estudiante quiera mejorar su nota (interrogación, tarea o proyecto). Importante esta actividad opcional **no puede ser entregada con atraso** ya que las votaciones comienzan el día siguiente a la fecha de entrega de la tarea.

Entregables

- Por cada actividad, usted debe diseñar y codificar al menos 3 ejemplos (como se vio en clase) para probar que las reglas y transformaciones solicitadas.
 - Los ejemplos deben ser diferentes.
 - Deben demostrar los casos que consideraron para implementar.
 - Por ejemplo, para `IfWithoutElse` transformation, pueden probar con ifs con varias líneas en el if-body, ifs que si tienen else-body y, por lo tanto, la transformación no aplica, ifs donde su if-body tenga una sola línea, etc.
- Debe subir el código de su tarea.
- Se debe subir la url a un vídeo corto de **máximo 4 minutos** donde muestre que su código funciona. Por cada transformación o warning debe:
 - Explicar rápidamente la lógica de su implementación.
 - Explicar los ejemplos con los que probaron la regla.
 - Ejecutar y mostrar que su código detecta los warnings o aplica las transformaciones efectivamente.
- Los que realicen la actividad 3 deben subir la url de un vídeo separado explicando solo la regla o transformación desarrollada en dicha actividad. El vídeo debe ser de **un minuto**.

Archivos iniciales

En canvas se encuentra el código desarrollado por el profesor en clase con los siguientes archivos:

- *input code (folder)* – que contiene el código utilizado en clases para probar las reglas de detección y transformación.
- *rewriter (folder)* – que contiene las reglas de transformación implementadas en clase.
- *rules (folders)* – que contiene las reglas de detección de code-smells implementadas en clase.
- *trasnformed code (folder)* – contiene el resultado de aplicar las reglas de transformación sobre el código de entrada (input code).
- *analyze.py* – ejecuta todas las reglas de detección de code-smells sobre el código de entrada (input code).
- *transform.py* – ejecuta todas las reglas de detección de transformación sobre el código de entrada (input code).

Para la tarea usted debe modificar o agregar código a los archivos anteriormente mencionados según corresponda.

Reportar problemas en el equipo

En el caso de que algún integrante no aportara como fue esperado en la tarea, podrán reportarlo enviando un correo con asunto **Problema Equipo {NumeroGrupo} Testing** a juanandresarriagada@uc.cl con copia a juanpablo.sandoval@uc.cl explicando en detalle lo ocurrido. Posterior a eso se revisara el caso en detalle con los involucrados y se analizara si corresponde aplicar algún descuento. Instamos a todas las parejas que mantengan una buena comunicación y sean responsables con el resto de su equipo para evitar problemas de este estilo.

Restricciones y alcances

- Su programa debe ser desarrollado en **Python 3.9 o superior**.
- Los archivos de código entregados deben terminar con la extensión **.py**.
- En caso de dudas con respecto al enunciado deben realizarlas en un foro relacionado a la tarea que se encontrara disponible en canvas.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier librería de Python adicional a las utilizadas en el código base se encuentran prohibidas. En caso de que estimes necesario podrás preguntar en el foro de la tarea por el uso de alguna librería adicional.

Entrega

Código: Deberán entregar el código que utilizaron para su tarea junto con los ejemplos que utilizaron para probar su funcionamiento. La entrega se realizara por medio de un buzón de canvas habilitado para esta tarea.

Videos: Deben subir los videos a YouTube e incluir las url en un archivo .txt que entregaran en el buzón junto con el código. Recuerden que en caso de hacer la actividad 3 deberán subir un video separado al que entreguen para las actividades anteriores y este sera compartido a sus compañeros. Importante que señalen a que video corresponde cada url y que los videos no se excedan de la duración máxima establecida. Ambos videos deben estar en configuración **"No Listado"**, o sea, **que solo se puedan acceder desde un url**².

Atraso: Se efectuara un descuento por entregar tareas atrasadas. Se descontara 0.5 si la tarea se entrega con menos de una hora de retraso. El puntaje final en caso de atraso seria calculado mediante la siguiente formula:

$$PuntajeFinal = PuntajeObtenido - (0,5 + 0,05 \cdot k)$$

, donde k es el numero de horas de retraso menos uno.

Integridad académica

Este curso se adscribe al Código de Honor establecido por la Escuela de Ingeniería. Todo trabajo evaluado en este curso debe ser hecho **individualmente** o en **los grupos asignados** según sea definido en la evaluación y **sin apoyo de terceros**. Se espera que los alumnos mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería.

²Con el fin de poder resguardar la integridad académica

¡Éxito! :)